



# **Model Driven Engineering in Practice**

Doctoral Theses

**Medve Anna**

MSC in Computer Science and Mathematics

Eötvös Loránd University PhD School of Computer Science  
Basics and Methodology of Informatics

Leader of PhD School:  
Dr. Benczúr András, professor

Leader of doctoral program:  
Dr. Demetrovics János, academician

Supervisor:  
Dr. Kozma László, professor

Research Laboratory:  
Department of Software Technology And Methodology

This research was supported by TÁMOP4.2.1./B-09/1/KMR-2010-0003, Pannontej Ltd., and Sonepar Ltd.

2013

## Introduction

The integration of new requirements in the software process always means hidden risks with respect to the correctness and costs of software change and of configuration management. Using the model driven technology [OMG12], even at high levels of abstraction, it is possible to manage the interoperability of the elements of software systems in the case of data evolutions as well as interactions on operation levels, even in the case of large complex systems.

In order to effectively use model driven technology, we need to include formal methods as well as to evolve from a code-centred development to a model-centred development, highlighting modelling and model transformations, architectural design and integration, and model correctness [Kup01, Béz04, Koz04, Hoo08, Jéz08, Len08, Cha09, Cla09, Ben10, Ist10, Pat10, Bra12, Jéz12].

Model Driven Architecture (MDA) [OMG12] is the product of Object Management Group (OMG). MDA is a framework containing the collection of principles and regulations in order to create model driven methodologies to structure software processes and software products. With the MDA methodology [OMG03] software products get separated with multilevel models above the code and the platform in a formal system model independent from implementation technologies which, in a separate implementation, can be executed on several possible platforms with multilevel model transformations.

Model Driven Engineering (MDE) is a model driven development methodology based on domain-specific languages [Obe10] and using the principles of MDA [OMG03].

The *MDE principles are*: model-based representation, aspect modelling, reuse, and automation.

## Objectives

The subject of our research is the process of model driven software engineering and the practical application of model driven technologies. We focus on the main MDE principles: *aspect viewpoint* with MDA architecture, *reuse* with software design patterns and software components, and *domain-specific modelling technologies*. The state of automation of these tools significantly determines the abilities of the potential products, the possibilities of their reuse, the set of modelling techniques, and their order in the process.

Due to the state of automation and reusability the development is executed with typically complex technologies using the “one problem – one modelling language” approach for each of the specific MDE aspects, which are then integrated by the principles for of model

integration. Due to these technological features the implementation of MDE into practice depends on the organisational capabilities.

In the model driven process, the inputs of the transformation techniques are: the business, the architectural and design patterns, the technology choice, as well as the quality requirements with the corresponding observation points [OMG03]. Several types of model driven architectural patterns can be followed, based on the available domain-specific knowledge bases and according to the extent of technology dependence on the quality requirements. These result in that business models are sharply embedded in platform models and requirement developments cannot be closed down preceding decisions about implementation technologies; but even during requirement development performance capability must be examined and test cases must be developed in order to increase quality. In the process of using a domain-specific knowledge base there is no sharp separation between the computational independent architectural layer and the platform independent architectural layer, and usually platform models are available too. In a process like this a high degree of automation is necessary.

Our work focuses on the practical application of MDE principles.

We look for the possibilities for the effective use of software design patterns for insuring modelling continuity in the model driven process and we also consider the key role of the architectural model in this process [Per92]. We investigate several possibilities for the application of MDE technologies in business process modelling for determining and regulating business process properties. This leads to the elimination of the most frequent errors of requirements elicitation and analysis. Reuse and aspect management are greatly enhanced by standard based development of requirements. Efficiency is increased by model-based testing and requirement-generated test cases. We examine the possibility of the organisation of the technological line.

### **Structure of the dissertation**

The dissertation is composed of an introduction, four numbered chapters, and a summary. The introduction contains the motivation and the outline of the 3 theses of the dissertation. In the first chapter we present the basic notions of model driven engineering and we outline the dimensions of model driven software development and the technologies investigated in the rest of the dissertation. The three theses of the dissertation are presented as follows: in Chapter 2 Thesis 1 provides a scheme to create a technological line for MDE, and the case of instantiation with ITU-T System Design Languages then to extend with Verimag IFx 2.0 Toolbox verification technology. In Chapter 3 Thesis 2 provides a methodology to model and

to reuse software design patterns (architectural, design, and domain-specific) woven at different abstraction levels within the MDE process. In Chapter 4 Thesis 3 provides a methodology for business process change management with MDE process based on User Requirements Notations [ITU12]. The core is the business process modelling in cooperative work of business actors who estimate the parameters values of their processes. We elaborated this methodology at the request of several enterprises of Lake Balaton Region.

The introductory part of each chapter consists of a survey of the research related to our work and the motivation. In the summary we present further research directions.

### **Applied methods**

The main methods of our research are: review, and reference of relevant publications (especially survey and classification papers); experimental observation and testing of relevant automated engineering environments; as well as development of our own methodologies and using them in concrete industrial projects for domain-specific process modelling and rapid application developments.

### **Presentation of our results**

#### **Thesis 1: Technological line for model driven engineering**

**Thesis 1** We provide a scheme to create a technological line for Model Driven Engineering (MDE) and integrated formal verification. **(Chapter 2)**

1.1. Set-up of a scheme to create technological lines for MDE by using the formal language family—*ITU-T System Design Languages* developed by *International Telecommunication Union*.

1.2. The augmentation of the scheme presented in thesis 1.1 with known verification techniques and tools from *the related literature and practice*.

In model driven processes the use of domain-specific modelling languages is indispensable. There is also a need for verification tools integrated in model driven technologies and frameworks which have an important role in specification languages and tools in the early phase of development. Our aim is to provide the users with a recommendation for practical model driven software engineering from design to code. For this it is necessary to find relations among the technological elements, to support the validation of the model where it is possible and to achieve automatic model transformation as far as code generation [Med08b].

In *Thesis 1.1* we set up a scheme for the elements of technological lines for MDE. The variables in the scheme of technological lines are as follows: domain-specific modelling languages, the unified modelling language, programming languages, software components and the standards. In the organisation of technological lines we highlight the formal methods and the standards of other language technologies [Med05, Med08b, Lei07].

We provide the instantiation of the principal technological line scheme with the ITU-T systems engineering languages (International Telecommunication Union (ITU) [ITU12] which cover the tasks of the entire software process. ITU-T standardisation is controlled and it is harmonised with the OMG specifications. It is also wide-spread in various fields of industry, in the area of medical devices development and in other scientific domains.

We represent the place and relations of the languages in a dependency graph in the software process in order to analyse the relations among various formal methods. We provide two technological line specifications by combining ITU-T engineering languages and UML and DSML languages whose variants we sampled onto 12 lines relevant for the processes of development, reengineering and reverse engineering. We recommend the User Requirements Notation [ITU12] standard-based requirements engineering and conceptional design.

In *Thesis 1.2* we make recommendation for the augmentation of the technological lines with existing verification technologies, as IFx 2.0 Toolbox, the TTCN-3 automated testing tool and the UML-TP principal toolbox into model-based testing [IFx12, Boz4, ITU12, OMG12].

We provide a method and choose known technologies in experimental ways in order to combine static analysers and automatic model checkers to formally verify the models during analysis and design. For this we made experiments with the integration of the verification tools in the software process [Tót07]. We investigate [Bor10] the application of the automatic model checkers built in commercial software for analyses and model transformations and their combination [Orb08] with IFx toolbox [Orb08].

We elaborate the process of integration of the IFx 2.0 Toolbox in the modelling process for accessing its verification services as well as the insertion of C++ language components into the set of models under verification. We outline the job of verifiers as verification service and provide a case study for their activities [Med10b] with the one-time verification of the multilayer transformation of PIM models [Med11b]. We generalize these experiments for the selection of the verification toolbox and the organisation of the verification process [Med12b].

*The publications of Thesis 1: Med08b, Med10b, Med11b, Med12b. Our further publications in this topic: Med05, Med07c, Lei07, Bor10, Tót07, Orb08.*

## Thesis 2: Application of Software Design Patterns in MDE Process

**Thesis 2** We create a methodology for a *framework* and *reuse* by *weaving software design patterns* into model driven engineering. **(Chapter 3)**

2.1 We provide a methodology and a case study to *build and reuse a framework* by weaving *POSA2 architectural pattern languages* and *GOF design patterns* into UML-based MDE.

2.2 We provide a methodology and a case study to build and reuse a framework and pattern language by weaving *POSA2 architectural pattern languages*, *GOF program design patterns*, and *SDL Pattern message services patterns* into SDL-based MDE.

In **Thesis 2** we recommend and provide a procedure and a case study in order to create a pattern collection in a framework by applying software design patterns and to apply it as model product with reuse in order to improve and speed up the MDE process among and within the various levels of abstraction. We claim that this is feasible by using POSA architectural pattern language because MDE architectural decisions are taken on the basis of the platform models of the known domain-specific implementations and on the basis of the requirements generated from the generic domain-specific models that are architecture relevant [omg03]. We presented this relation of model creation in relation to MDA model transformation and the tasks of model checking in [Med10b].

For the general as well as for the domain specific engineering we applied the architectural pattern language of distributed systems called Pattern-Oriented Software Architecture for Concurrent and Networked Objects , shortly POSA2 [Sch00, Bus96]. We apply the program design patterns, named GOF Design Patterns [Gam96], for the modelling of the behaviour of POSA2 architectural patterns where possible. The applied domain-specific patterns are SDL Pattern Pool patterns for message services management [Got03].

In the use of software patterns the emphasis is placed upon requirements engineering and we concentrate on structural and organisational elements from which we get to further abstraction levels by modelling and applying patterns. In the wide-spread pattern-based methods implementation is highlighted rather than modelling.

In **Thesis 2.1** respectively **Thesis 2.2** we provide methodologies and cases studies to create and reuse model collections of architectural patterns in MDE for UML development [Med10]

respectively for SDL development too [Med08]. The case study of the reuse is about system context design and rapid development by modelling (Chapters 3.2.2, 3.2.3, 3.3.5 in Dissertation); it is also an example of traceability among the technologies in Thesis 1.

In **Thesis 2.3** we show that we can develop a domain-specific pattern language for rapid development by modelling with domain-specific language the architectural patterns for distributed systems and object-oriented design patterns. In the detailed design we can extend the obtained pattern language by applying domain-specific design patterns during which we can separate further abstraction levels for the modelling of quality requirements and thus for the increase of testability and traceability. The technologies used to create the SDL Macro-pattern language are: SDL, MSC, POSA2, GOF, and SDL Pattern Pool [Med06, Med08].

This method can be used for any domain by specialisation of the client-server communication model of the patterns. The analysis gets simplified to the correspondences on notion level and the specialisation of generic models along the subsystem's functions, even beginner engineers can execute pattern application.

*The publications of Thesis 2: Med08, Med10. Our further publications: Med06, Med07b, Med07c.*

### **Thesis 3: Business Process Change Management with MDE**

**Thesis 3** We develop a methodology for *business process change management* with MDE. **(Chapter 4)**

3.1. We provide a scheme to create a *technological line for MDE of the business domain* with the ITU-T System Design Languages standardized in telecommunication domain built on the technologies in Thesis 1.1 and 1.2.

3.2. We provide a methodology supported by industrial application for the specification of *business processes and the implementation of inspection with cooperative work organization* with business process owners, business analysts, quality experts and model developers built on the technologies in Thesis 3.1.

3.3. We provide a methodology supported by industrial application built on the technologies in Thesis 3.1 for the creation of a *framework for change management* with the alignment of business processes and strategies managed by the concerned parties.

Model driven technologies provide many advantages for the design and maintenance of IT systems by supporting aspects and reuse. This is advantageous in the business domain where

it is typical that the organisational structure is in constant development along the business strategy and the software systems meet the everyday demands of several concerned target groups. Today business intelligence in software is an emphasised part of enterprise property. For this reason, during both the modernisation of IT systems and the management of business changes software engineering projects mostly alter the existent software system and seldom replace it completely [Min98]. For these system aspects, model driven requirements engineering [Buh98, Liu01, Wei05, Lam09, Amy11, Mus12] provides satisfactory support and a framework for change management for which we make recommendation in *Thesis 3*. We follow the goal-oriented and risk-adapted methods along which business and IT views can be correlated. In *Thesis 3.1* we provide a scheme for the organisation of the technology line for the MDE in business domain with ITU-T technologies [Med07, Med8b].

In order to redesign business strategies and applications it is required to specify the recent and future system state. In *Thesis 3.2* for the cooperative exploration and regulation of business processes we provide a generic methodology on URN technologies. The method enables business process owners and quality managers to estimate the values of their process parameters. Based on the attributes elicited in the process specification template the developers model the business processes and then the business analysts in charge of enterprise departments and the process owners apply inspection techniques for the process specifications to validate the requirements given in their process template. We elaborated this method for a concrete industrial demand. It is generic due to the use of MDE [Óvá07, Med10c, Med11d].

Company systems are typically critical systems: they depend on human factors and management methods. In *Thesis 3.3* we provide a URN-based methodology with the coordinated development of goals and scenario models for the development of business processes and strategies in order to create, reuse, and specialise a generic framework. They are used by starting from goal models to specialise them in several iterations [Med12, Med12c]. As part of the framework we provide generic goal-oriented models for the chapters of the information security standard [ISO12] and its specialisation for e-commerce and its reuse for concrete reengineering at enterprises [Med09, Med11, Med11c]. We elaborate patterns and a case study to integrate mobile applications in enterprise technology by the applications of URN, UML, and MSC technologies by starting from the use cases descriptions [Med12c].

*The Publications of Thesis 3.: Med07, Med10c, Med11, Med11c, Med11d, Med12, Med12c. Our further Publications in the Topic: Óvá07, Med09.*



## Basic Publications of the Theses

- Med07 A. Medve: *Enterprise modelling with the joint use of User Requirements Notation (URN) and UML*. Book chapter In: P.Rittgen (Ed.), *Enterprise Modelling and Computation with UML*, Hershey, USA. Development Ed. IDEA Group, 46-69, 2007.
- Med08 A.Medve, Il.Ober: *From Models to Components: Filling the Gap with SDL Macro-patterns*, Journal of Computer Standards & Interfaces, Elsevier, 30 (5), 315-322, 2008.
- Med10 A.Medve, L.Kozma, Il.Ober: General Modelling Approach Based on the Intensive Use of Architectural and Design Patterns, In: H.Weghom, P.Isaias, R.Vasiu (Eds): *Proc. of the IADIS Int. Conf. Applied Computing 2010*, pp.251-256, Timisoara, Romania, 2010.
- Med10b A.Medve, Gy.Orbán, L.Kozma: Let's Go the Verification Engineering, In L.Varga, G.Weiss, G.Kusper (Eds.) *Proc. of the 8th Int. Conf. on Applied Informatic (ICAI 2010)*, pp. 417-428, Eger, Hungary, 2010.
- Med10c Medve,A.: ProjektProcess módszertan fejlesztése - Pannontej Zrt., *Kutatásjelentés*, PE-MIK-VIRT-2/10, 2010.
- Med11 A.Medve: Standards-based Framework for Functionally Integrated Engineering of Information Systems. In T.Szmuc, M.Biró (Eds.) *5th IFIP TC2 Central and Eastern European Conf. on Software Engineering Techniques (CEE-SET 2011)*, pp.52-58, Debrecen, Hungary, 2011.
- Med11b A.Medve, L.Kozma: MDA Process and Model Refinements Based on Verimag IFxVerification Tools, In: H.F.Pop, A.Bege (Eds.) *Proc. 8th Joint Conf. on Mathematics and Computer Science, MaCS 2010*, Komarno, Slovakia, July14-17, 2010, Selected Papers, Novadat Ltd., pp. 323-336, 2011.
- Med11c Medve,A.: Módszertan elektronikus üzleti technológiák integrálására, Sonepar Magyarország Kft. B2B&B2C&URN projekt, *Kutatásjelentés*, PE-MIK-VIRT-3/11, 2011.
- Med11d Medve,A.: UCM2MSC minőség audit támogatás a ProjektProcessz folyamatspecifikációs módszertanban, Pannontej Zrt., *Kutatásjelentés*, PE-MIK-VIRT-1/11, 2011.
- Med12 A.Medve: Model-based Framework for Integrated Evolution of Business and IT Changes. In S.Hammoudi, M.Sinderen, J.Cordeiro (Eds.) *ICSOF 2012. Proc. of the 7th Int. Conf. on Software Paradigm Trends*, 24 - 27 July 2012, Rome, Italy, pp. 255-260, SciTePress, 2012.
- Med12b A.Medve: *Towards MDE Improvements from Integrated Formal Verifications*, J. STUDIA Univ. Babes-Bolyai, Vol. LVII, No. 1, pp. 89-104, 2012.
- Med12c Medve,A. Augmented Enterprise Framework, Mobil informatikai szolgáltatások üzleti folyamatainak modellvezérelt fejlesztése. Informatix Zrt. AEF projekt, *Kutatásjelentés*, PE-MIK-VIRT-2/12, 2012.

## Other relevant publications in the topic of dissertation

- Bor10 Zs.Borsi, L.Kozma, A.Medve: Component-based Software Engineering Verification Problems, L.Varga, G.Weiss, G.ábor Kusper (Eds), *Proc.of the 8th Int. Conf. on Applied Informatics (ICAI 2010)* Eger, Hungary, January 27-30, 2010, pp. 391-400, 2010.
- Lei07 F.Leitold, A.Medve, L.Kovács: SIP security problems in NGM Services, In: K. Al-Begain (Ed.) *Int. Conf. on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007)*, Cardiff, UK, pp. 241-246, IEEE, 2007.
- Med05 A.Medve: Standardized Formal Languages for Reliable Model Engineering, *Proc. 25rd. microCAD Int. Conf. on Applied Information Engineering*, Vol. N, pp. 315-321, ITTC, Miskolc, Hungary, 2005.
- Med06 A.Medve, Il.Ober: Design patterns in SDL: The component configuration pattern case study, *Kutatásjelentés*, PE-MIK-IRT-3/06, 2006.
- Med07b Medve,A., Szima,I.: Esettanulmány a POSA2 Szolgáltatáselérés mintanyelv implementálására, *Kutatásjelentés*, PE-MIK-IRT-5/07, 2007.
- Med07c Medve, A.:UML és SDL társulása a valós idejű és egymásba ágyazott rendszerek modellezésére, Magyar-Francia Tét F/16, *Kutatásjelentés*, PE-MIK-IRT-2/07, 2007.
- Óva07 Óvádi,P., Medve,A.: SAP PM bevezetése - Le Belier Magyarország, *Kutatásjelentés*, IRT2/2007.
- Tót07 Tóth,L., Medve,A.: VIATRA kísérlet, *Kutatásjelentés*, PE-MIK-IRT-3/07, 2007.
- Szi08 Szima,I., Medve Anna: Medve,A., Szima,I.: Esettanulmány a POSA2 Szolgáltatáselérés mintanyelv implementálására, *Kutatásjelentés*, PE-MIK-IRT-5/07, 2007.
- Orb08 Orbán György, Medve Anna: IFx-CADP-SPIN kísérlet, *Kutatásjelentés*, PE-MIK-IRT-3/08, 2008.
- Med09 Medve,A., Nagy,T., Nagy,R., Nagy,N.: Kísérlet az ISO/IEC 27001 és 27002 modellezésére és az átfedések kezelésére, *Kutatásjelentés*, PE-MIK-VIRT-1/09, 2009.

## References

- Amy11 D.Amyot, G.Mussbacher: User Requirements Notation: The First Ten Years, The Next Ten Years (Invited paper), *Journal of Software*, 6(5):747-768, Academy Publisher, 2011.
- Ben10 R.Bendraou, J.-M.Jézéquel, M.-P.Gervais, X.Blanc: A Comparison of Six UML-Based Languages for Software Process Modeling. *IEEE Trans. Software Eng.* 36(5): 662-675, 2010.
- Béz04 J.Bézivin: Model Engineering for Software Modernization, In: 11th Working Conf. on Reverse Engineering (WCRE 2004), Delft, The Netherlands. IEEE Computer Society, 2004.
- Boz04 M.Bozga, S.Graf, L.Mounier, Il.Ober, Iu.Ober, J.Sifakis: The IF toolset, Formal Methods for the Design of Real-Time Systems, LNCS 3185, p. 237-267, 2004.
- Bra12 M.Brambilla, J.Cabot, M.Wimmer: Model-Driven Software Engineering in Practice. Morgan, 2012.
- Buh98 R.J.A.Buhr: Use Case Maps as Architectural Entities for Complex Systems, In: *IEEE Transactions on Software Engineering*, 24(12):1131-1155, 1998.
- Bus96 F.Buschmann, R.Meunier, H.Rohnert, P.Sommerlad: Pattern-Oriented Software Architecture, Volume 1: A System of Patterns. John Wiley & Sons, 1996.
- Cha11 A.W.H.Chang, G.Adamis, L.Erös, G.Kovács, T.Csöndes: A New Approach in Model-Based Testing: Designing Test Models in TTCN-3. In: Il.Ober, Iu.Ober (Eds.) *SDL Forum Toulouse*, pp.90-105, 2011.
- Cla09 E.M.Clark, E.A.Emerson, J.Sifakis: Model checking: algorithmic verification and debugging, *Communications of the ACM* 52(11):74-84, 2009.
- Gam95 E.Gamm, R.Helm, R.Johnson, J.Vlissides: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, 1995.
- Got03 R.Gotzhein: Consolidating and Applying the SDL-Pattern Approach: A Detailed Case Study, *Information and Software Technology*, 45(11):727-741, Elsevier, 2003.
- Hoo08 J.Hooman, H.Kugler, I.Ober, Y.Yushtein: Supporting UML-based Development of Embedded Systems by Formal Techniques, *Int. Journal of Software and Systems Modeling*, 7(2):131-155, Springer, 2008.
- Hut11 J.Hutchinson, M.Rouncefield, J.Whittle: Model-driven engineering practices in industry. In R. Taylor et al.(Eds.): *Proc. of the 33rd Int. Conf. on Software Engineering, ICSE 2011, ACM*, p. 633-642, 2011.
- Ist10 Z.Istene: Formal Methods in Robot Control Systems L.Varga, G.Weiss, G.ábor Kúspér (Eds.), *Proceedings of the 8th Int. Conf. on Applied Informatics in Eger, Hungary, January 27-30, 2010*, pp. 391-400, 2010.
- IFx12 IFx-OMEGA Toolset: <http://www.irit.fr/ifx/>; <http://www-if.imag.fr/>; <http://www-omega.imag.fr/>, 2012.
- ISO12 ISO/IEC 27001, ISO/IEC 27002 Information Security Standard, <http://www.27000.org>, 2012.
- ITU12 ITU-T Recommendation Z. Series: Languages and General Software Aspects for Telecommunication Systems, <http://www.itu.int/rec/T-REC-z>.
- Jéz08 J.-M.Jézéquel: Model driven design and aspect weaving. *International Journal of Software and Systems Modeling* 7(2):209-218, Springer, 2008.
- Jéz12 J.M.Jézéquel, B.Combemale, D.Vojtisek: *Ingénierie Dirigée par les Modèles: des concepts à la pratique*, Ellipses, Paris, 2012.
- Koz04 Kozma,L., Varga,L.: *A szoftvertechnológia elméleti kérdései*, ELTE Eötvös Kiadó, Budapest, 2004.
- Kup01 O.Kupferman, M.Y.Vardi, P.Wolper: Module checking. *J. Information and Computation* 164(2):322-344, Elsevier, 2001.
- Lam09 A.-van-Lamsweerde: Requirements Engineering – From System Goals to UML Models to Software Specifications. Wiley, 2009.
- Len08 L.Lengyel, T.Levendovszky, H.Charaf: Validated model transformation-driven software development, *International Journal of Computer Applications in Technology*, 31(1):106-119, 2008.
- Liu01 L.Liu, E.Yu: From Requirements to Architectural Design - Using Goals and Scenarios, *STRAW'2001, Toronto*, pp.1- 9, <http://www.cin.ufpe.br/~straw01>, 2001.
- Min98 H.Mintzberg: *Structure et dynamique des organisations*. Edition d'Organisation, Paris, 1998.
- Mus12 G.Mussbacher, J.Araújo, A.Moreira, D.Amyot: AoURN-based Modeling and Analysis of Software Product Lines, *Software Quality Journal*, 20(3-4):645-687, 2012.
- Obe10 I.Ober, L.Féraud, C.Percebois: Dealing with variability within a family of domain-specific languages: comparative analysis of different techniques. *Innovations in Systems and SE* 6(1-2):21-28, 2010.
- OMG03 OMG Strategy Study Group MDA Guide Version 1.0.1, 2003, <http://www.omg.org/mda>, 2012.
- OMG12 OMG Specifications – [http://www.omg.org/technology/documents/spec\\_catalog.htm](http://www.omg.org/technology/documents/spec_catalog.htm);
- Pat10 A.Pataricza. et al.: Workflow-driven Tool Integration Using Model Transformations, In: G.Engels (Ed.) *Graph Transformations and Model Driven Engineering*, pp- 224-248, LNCS 5765, Springer, 2010.
- Per92 D.E.Perry, A.L.Wolf: Foundations for the study of Software Architecture. *Software Engineering Notes ACM SIGSOFT*, 17(4):40-42, 1992.
- Sch00 D.C.Schmidt, M.Stal, H.Rohnert, F.Buschmann: *Pattern-Oriented Software Architecture., Patterns for Concurrent and Networked Objects* Wiley, (<http://www.cs.wustl.edu/~schmidt/POSA/>) 2000.
- Wei05 M.Weiss, D.Amyot: Business Process Modeling with URN. *J. E-Business Research*, 1(3):63-90, 2005.